Вариант 1

```
@echo off
set todir=%1
if "%todir%"=="" goto intro
:getfile
shift
if "%1"=="" goto end
copy /B %1 %todir% >nul
goto getfile
:intro
echo Копирует произвольное количество
echo файлов в каталог
echo Вызов:
echo   %0 directory file1 file2 ...
exit /B
:end
echo Копирование закончено
```

Вариант 2

```
@echo off
if "%1"=="" (
  echo Копирует произвольное количество
  echo файлов в указанный каталог
  echo Вызов:
  echo  %~NX0 directory file1 file2 ...
  exit /B
)
:getfile
if not "%2"=="" (
  copy /B %2 %1 >nul
  shift /2
  goto getfile
)
echo Копирование закончено
```

1

```
@echo off
rem  проверяет существует ли каталог %1, и, если объект с
rem  таким именем существует, то проверяет, файл это или каталог
if not exist %1 (
  echo %1 : no such file or directory
) else (
  dir /A:D %1 >nul
  if errorlevel 1 (
    echo %1 directory does not exist, but %1 file exists
  ) else (
    echo %1 directory exists
  )
)
```

2

```
@echo off
rem  проверяет, существует ли переменная окружения
rem  STDOUT_REDIRECTED, и, если нет, то определяет её

if not defined STDOUT_REDIRECTED  (
  rem вариант: if "%STDOUT_REDIRECTED%"==""
  set STDOUT_REDIRECTED=yes
  cmd.exe /c %0 %* >%1
  rem вариант: call %0 %* >%1
  exit /b %ERRORLEVEL%
)

rem здесь символы < и > экранированы
echo 2 x 2 = 4
echo 3 x 3 ^< 10
echo 4 x 4 ^> 15
```

3

Вариант 1

```
@echo off
rem Сцепляет все указанные файлы
rem (группы файлов) - * и ? допустимы
rem вместе в один указанный файл.

if "%2"=="" goto USAGE
set RES=%1
shift

:NEXT
for %%i in (%1) do call add %%i %RES%
shift
if "%1"=="" goto END
goto NEXT

:USAGE
echo Usage: %0 destination files

:END
echo on
```

Вариант 2

```
@echo off
rem Сцепляет все указанные файлы
rem (группы файлов) - * и ? допустимы
rem вместе в один указанный файл.

if "%2"=="" (
  echo Usage: %~nx0 destination files
  goto END
) else (
  :NEXT
  for %%i in (%2) do call add %%i %1
  shift /2
  if not "%2"=="" goto NEXT
)

:END
echo on
```

CONCAT.BAT

```
if not exist %2 goto create
copy /B %2+%1 >nul
goto exit
:create
copy /B %1 %2 >nul
:exit
```

```
if exist %2 (
  copy /B %2+%1 >nul
) ELSE (
  copy /B %1 %2 >nul
)
```

ADD.BAT

4

```
@echo off
rem rename all *.htm  in %1 and all
rem it's subdirectories onto *.html
for /R %1 %%i in (*.htm) do (
  echo rename %%I
  ren "%%i" "%%~ni.html"
)
```

5

```
@echo off
rem Выводит все файлы и подкаталоги текущего каталога
for /R %%i in (.) do (
  echo directory %%i
  dir "%%i\*.*"
  pause
)
```

6

```
@echo off
rem Вычисляет факториал:
rem "функция" - call :метка параметры

setlocal enabledelayedexpansion

call :factorial %1
echo %RES%
exit /b

rem function factorial n

:factorial

if %1 == 0  (
  set /a RES = 1
) else (
  set /a  N = %1 - 1
  call :factorial !N!
  set /a RES *= %1
)
exit /B
```

7

```
@echo off

:start
  set /A X = %RANDOM%*%1/32768 + 1
if %time:~-2% GTR 3 goto start

echo %X%
```

8

```
@echo off
rem Проверяет, есть ли поддержка extensions

verify LALALA 2>nul
rem В старых версиях CMD.EXE команда SETLOCAL не
rem изменяет значения ERRORLEVEL. Команда verify
rem с некорректным параметром нужна здесь именно
rem для того, чтобы установить ERRORLEVEL в не 0

setlocal ENABLEEXTENSIONS 2>nul
if not ERRORLEVEL 1 (
  echo Extensions are enabled.
) else (
  echo Unable to enable extensions.
)
```

9

```
@echo off
rem вычисляет степень 2 (до 31-й включительно)
set /P n="Enter power "
if %n% GEQ 0 goto nonnegative
  echo Incorrect input: power should be a non-negative
  exit /B
:nonnegative
if %n% LSS 32 goto okpower
  echo Incorrect input: power is too big
  exit /B
:okpower
  rem вариант: set /A b =1^<^<%n%
  set /A b =1"<<"%n%
  echo 2^^%n% = %b%
```

10A

```
@echo off
rem вычисляет степень 2 (до 31-й включительно)
rem Включаем отложенное вычисление переменных
setlocal enabledelayedexpansion
set /P n="Enter power "
if %n% LSS 0 (
  echo Incorrect input: power should be a non-negative
) else (
if %n% GEQ 32 (
  echo Incorrect input: power is too big
) else (
  rem  set /A b =1^<^<%n%
  set /A b =1"<<"%n%
  echo 2**%n% = !b!
)
```

10B

```
@echo off
rem Выводит список всех environment-переменных, пронумеровывая их.
rem Включаем отложенное вычисление переменных
setlocal enabledelayedexpansion
rem В противном случае в блоке (...) подстановка значения
rem переменной num произойдет до того, как она будет изменена
rem в процессе выполнения этого блока. Иначе говоря, в строке
rem       set /a num = %num% + 1
rem будет подставлен 0 - то значение, которое переменная num
rem имела на момент начала выполнения данного for-блока.
rem
rem Ограничители переменной ! ! вместо % % и указывают на то,
rem что подстановка значения переменной должна выполняться
rem только в момент выполнения соответствующей команды.
set /a num = 0
FOR /F "usebackq delims==" %%v IN (`set`) DO (
  set /a num = !num! + 1
  if !num! LSS 10 (
    echo 0!num!. %%v
  )  else (
    echo !num!. %%v
  )
)
```

11

```
@echo off

rem Читаем настройки из файла settings.txt, который располагается
rem в том же каталоге, что и bat-файл. Если не удалось распарсить
rem настройки - выходим с ненулевым кодом возврата.

call :read_settings %~sdp0settings.txt || exit /b 1

rem Прочитанные настройки:
echo MAIN
echo Build mode   : %BUILDMODE%
echo Compiler     : %COMPILER%
echo Architecture : %ARCH%
echo Options      : %OPTIONS%

rem Выход. Дальше - только функции.
exit /b 0
```

```
# Простой файл настроек
# Режим сборки
buildmode=release
# Компилятор
compiler=cl.exe
# Архитектура
arch=x86
```

settings.txt

```
:read_settings

rem Функция для чтения настроек из файла.
rem Вход:  %1 - имя файла с настройками

set SETTINGSFILE=%1

rem Проверка существования файла

if not exist %SETTINGSFILE% (
    echo FAIL: No such file: %SETTINGSFILE%
    exit /b 1
)

rem Обработка файла с настройками
rem Здесь:
rem   eol=# указывает на то, что содержимое строки, начиная с
rem   символа # и до ее конца, может быть пропущено как комментарий.
rem
rem   delims==  указывает, что, кроме пробела,
rem             разделителем значений будет и символ =
rem
rem   tokens=1,2 приводит к тому, что в переменную %%i будет
rem              занесен первый токен, а в %%j - второй.
rem

for /f "eol=# tokens=1,2 delims== " %%i in (%SETTINGSFILE%) do (
    rem В переменной i - ключ
    rem В переменной j - значение
    rem Мы транслируем это в переменные окружения
    set %%i=%%j
)

exit /b 0
```

```
@echo off
rem Генерирует все подмножества заданного множества;
rem множество задается как битовая маска

set subset=%1
:next
rem  print subset
call :printsubset %subset%
rem  generate next subset
set /A subset=(subset - 1)^&%1
if %subset% NEQ 0 goto :next
echo 0:
pause
exit /b

rem function printsubset bitsubset
:printsubset
set mask=%1
set output=%1:
set count=0

:loop
if %mask%==0 (
  echo %output%
  exit /b
)
set /A count=count+1
set /A lastbit=%mask%^&1
if %lastbit% EQU 1  set output=%output% %count%
set /A mask=mask">>"1
goto :loop
```

13

```
@echo off
rem Печатает таблицу
rem умножения на a
set /p a=Enter number:
set k=1
:loop
  set /a b = %a% * %k%
  echo %a% * %k% = %b%
  set /a k = %k% + 1
if %k% LEQ 10 goto loop
```

14A

```
@echo off
rem Включаем отложенное вычисление переменных
setlocal enabledelayedexpansion
set /P n=Enter number:
cls
for /L %%k in (1,1,10) do (
rem в set /A можно писать просто n, а не %n%
  set /A  b = n*%%k
  echo %n% * %%k = !b!
)
```

14B

```
@echo off
setlocal enabledelayedexpansion
for %%f in (*.*) do (
  if not "%~f0"=="%%~ff" (
    set subdir=%%~xf
    if !subdir!.==. (
      set subdir="Empty extension"
    ) else (
      set subdir="!subdir:~1!"
    )
    if not exist !subdir! md !subdir!
    move "%%f" !subdir! >nul
  )
)
```

Перебирает все файлы в текущем каталоге, для каждого расширения файлов создает каталог с названием, равным этому расширению, и переносит в него все файлы с этим расширением. Сам себя не трогает.

15

```
@echo off
setlocal enabledelayedexpansion
for %%f in (log.*) do (
  for /f "tokens=1-3* delims=./ " %%s in (%%f) do (
    set /a num=%%s+%%t+%%u
    if !num! neq 0 (
      if %%u lss 100 (set yyyy=20%%u) else (set yyyy=%%u)
      set dd=%%s
      if %%s lss 10 (
        if not "%%s:~0,1"=="0" (set dd=0%%s)
      )
      set mm=%%t
      if !mm! lss 10 (
        if not "!mm:~0,1!"=="0" (set mm=0!mm!)
      )
      dir /A:D !yyyy! >nul
      if errorlevel 1 (md !yyyy!)
      if exist !yyyy!\!mm!.!dd! (
        echo %%v >>!yyyy!\!mm!.!dd!
      ) else (
        echo %%v >!yyyy!\!mm!.!dd!
      )
    )
  )
)
```

16

```
@echo off
set /p str=
:loop
for /F "usebackq tokens=1* delims= " %%c in ('%str%') do (
echo %%c
set str=%%d
)
if not "%str%"=="" goto loop
```

17

---

1. %0 directory file1 file2 ...
Копирует все указанные файлы (или группы файлов - * и ? допустимы) в указанный каталог.
a. в старомодной манере: без операторных скобок - только с goto, зато заводим "настоящую" environment-переменную
b. "по-новому": используются shift /n, преобразования параметров, операторные скобки

2. Проверяет, существует ли указанный файл/каталог и кто он - файл или каталог. Тема: команды if exist, if errorlevel, if-else.

3. Проверяет установку переменной окружения и, если её нет, заводит эту переменную. Тема: cmd /c vs call, экранирование символов < и > символом экранирования ^

4. %0 destination file1 file2 ...
Сцепляет все указанные файлы (или группы файлов - * и ? допустимы) вместе в один указанный файл. Тема: вызов с возвратом - call с передачей параметров, просто for, перенаправление "мусорного" вывода.
a. в старомодной манере: без операторных скобок - только с goto, зато заводим "настоящую" environment-переменную
b. "по-новому". Тема: используются shift /n, преобразования параметров, операторные скобки

5. Меняет расширение всех htm-файлов в каталоге %1 и всех его подкаталогах на html. Тема: for /R, преобразования имён файлов.

6. Выводит все файлы и подкаталоги текущего каталога. Тема: for /R, каталог . (точка).

7. Вычисляет факториал. Тема: "функция" - call :метка параметры, рекурсия, отложенная подстановка переменных.

8. "Randomize": %RANDOM%, %TIME%, преобразование строк - подстроки.

9. Проверяет, есть ли поддержка extensionsю Тема: errorlevel, перенаправление вывода ошибок (2>nul).

10. Вычисляет степень двойки. Тема: set /P, set /A, арифметическое сравнение, арифметические операции.
a. в старомодной манере: без else - только с goto.
b. "по-новому": используя else

11. Выводит список всех environment-переменных, пронумеровывая их. Тема: for /F, опция usebackq, парсинг вывода команды, отложенное вычисление переменных

12. Читаем настройки из файла settings.txt, расположенного в том же каталоге, что и bat-файл. Тема: for /f и её опции, вызов "функции" по метке, exit /b   errorlevel

13. Генерирует все подмножества заданного множества; множество задается как битовая маска. Тема: арифметика, в т.ч. битовая арифметика.

14. Печатаем таблицу умножения числа из %1 на 1..10. Тема: set /p, set /a
a. в старомодной манере: используя goto.
b. "по-новому": используя for /L и отложенное вычисление переменных.

15. Перебирает все файлы в текущем каталоге, для каждого расширения файлов создает каталог с названием, равным этому расширению, и переносит в него все файлы с этим расширением. Сам себя не трогает. Тема: парсинг имён файлов, цикл for in.

16. Парсит все файлы с именем log. Каждая строчка в таком файле начинается с даты в формате число, месяц, год (в такой последовательности), разделённые пробелами или точками или /. Создаётся файл мм.дд в каталоге гггг, в который выводится всё оставшееся содержимое строки. При необходимости создаётся и каталог гггг.

17. Считывает строку, разбивает её на части, разделённые пробелами, и выводит каждую часть в новой строке.